L2/23-073

ISO/IEC JTC1/SC2/WG2/IRG N2581−feedback

Date 2023−03−22

Source:	TianHeng Shen(沈天珩, aka CheonHyeong Sim)
Title:	About the Encoding Model on Han Ligatures
Status:	Individual Contribution on IRG #60
Action:	To be considered by IRG

Background

For WS2021−03670 which is considered as a ligature of 蟲 and 鳥, Eiso Chan suggests not to encode it as a separate CJK Unified Ideograph, but to use the OpenType Feature "ccmp" or "liga" to form a ligature automatically by the font with the existing characters.

Even if there were some ligatures already encoded as separate CJK Unified Ideographs, e.g. 坊, 㘒, et cetera, just as Eiso pointed out, hundreds even thousands of this kind of characters could be provided, encoding all of them separately seems prone to huge disaster for future IRG works. For this reason, I agree with Eiso's opinion not to encode it as a separate CJK Unified Ideograph, but, how to deal with it may become a problem.

Proposal

My proposal is to add a new character named **HAN LIGATURE STARTER** into the *Ideographic Symbols and Punctuation* block, possibly be at U+16FE5, with the script property as Hani. Following is the possible Unicode Data for the new character; and the reason for proposing to add this new character will appear in a subsequent post.

16FE5;HAN LIGATURE STARTER;So;0;L;;;;;N;;;;;

1. The necessity of using IDCs in the sequences

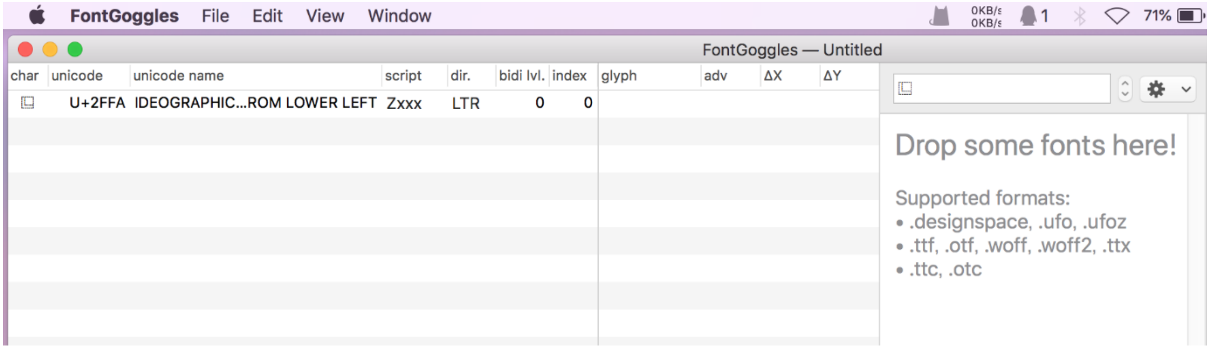
Among "ccmp" and "liga", I also agree with Eiso, i.e. the "ccmp" is preferred. But Eiso uses the sequence U+87F2(蟲) U+200D(ZWJ) U+9CE5(鳥) to express the ligature, which I strongly oppose. The problem here is, how could you know the ligature should be ⿱蟲鳥 but not ⿰蟲鳥 or the other structures? Till now we only find the former, but as a well developed encoding model, we must think about if one day in the future, the latter is found and need to be encoded, so how to deal with it? As a conclusion, the IDC characters have to be used in the sequences to show the unique structure between the components. Thus, the ZWJ will be no longer needed.

2. The necessity of using a starter

Firstly, it is widely known that, the usage of the IDSes today is limited to showing the structure of an unencoded character or a character difficult to show correctly on most devices, et cetera, but it is NOT used to form ligatures. Although Source Han Sans and Source Han Serif used the IDS to express 𰻝, 𰻞 and 𱁬 in the previous versions as an Easter Egg, it was just a temporary scheme before the release of CJK–ExtG.

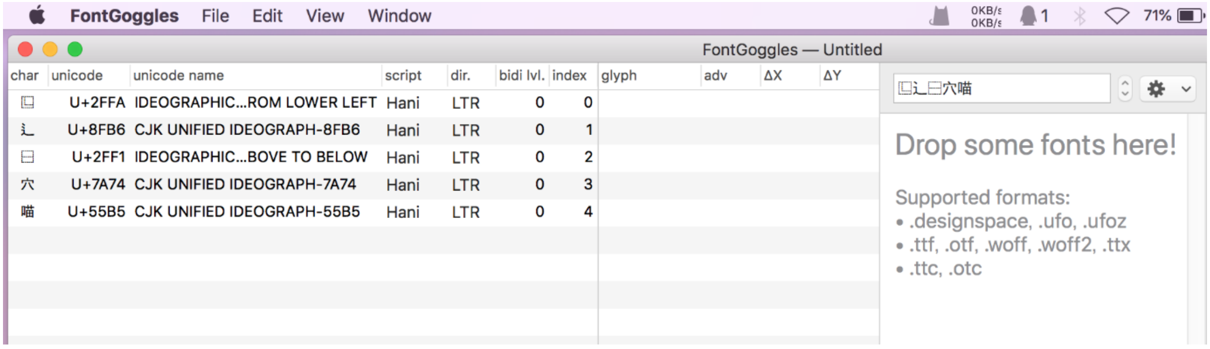
Secondly, an IDS always starts with an IDC, but the script property of the IDC characters are Zyyy(aka Common), which would lead to an error when forming the ligature in some cases.

Applying the OpenType Features has a prerequisite that, the characters in a sequence must have the same script property. If a character has the script property Zyyy(aka Common) or Zinh(aka Inherited), it will find the first previous character without the script property Zyyy or Zinh, and inherit its script property; if no such character found, then it will find the subsequent character and do the same thing; if still no such character found, then it will be Zxxx and apply the features under "script DFLT". Let us see some examples of the above explanations for easier understanding:



char	unicode	unicode name	script	dir.	bidi lvl.	index
⿰	U+2FFA	IDEOGRAPHIC...ROM LOWER LEFT	Zxxx	LTR	0	0

From the 1st picture, the IDC character appears independently, with neither any characters before it nor any characters after it, so it has nothing to inherit, then its script property fallbacks to Zxxx;



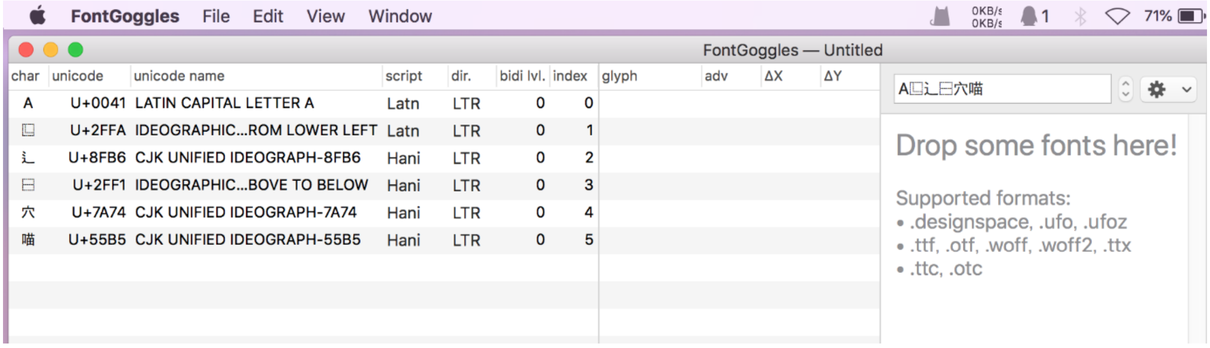
char	unicode	unicode name	script	dir.	bidi lvl.	index	glyph	adv	ΔX	ΔY
⿺	U+2FFA	IDEOGRAPHIC...ROM LOWER LEFT	Hani	LTR	0	0				
辶	U+8FB6	CJK UNIFIED IDEOGRAPH-8FB6	Hani	LTR	0	1				
⿱	U+2FF1	IDEOGRAPHIC...BOVE TO BELOW	Hani	LTR	0	2				
穴	U+7A74	CJK UNIFIED IDEOGRAPH-7A74	Hani	LTR	0	3				
喵	U+55B5	CJK UNIFIED IDEOGRAPH-55B5	Hani	LTR	0	4				

⿺辶⿱穴喵

Drop some fonts here!

Supported formats:
- .designspace, .ufo, .ufoz
- .ttf, .otf, .woff, .woff2, .ttx
- .ttc, .otc

From the 2nd picture, the IDC character U+2FFA does not have a character before it(obviously there will not be any character "without the script property Zyyy or Zinh" before it). So it inherits the script property of the character after it, i.e. Hani; the IDC character U+2FF1 do have a character without the script property Zyyy or Zinh before it, so it also inherits the script property Hani.



char	unicode	unicode name	script	dir.	bidi lvl.	index	glyph	adv	ΔX	ΔY
A	U+0041	LATIN CAPITAL LETTER A	Latn	LTR	0	0				
⿺	U+2FFA	IDEOGRAPHIC...ROM LOWER LEFT	Latn	LTR	0	1				
辶	U+8FB6	CJK UNIFIED IDEOGRAPH-8FB6	Hani	LTR	0	2				
⿱	U+2FF1	IDEOGRAPHIC...BOVE TO BELOW	Hani	LTR	0	3				
穴	U+7A74	CJK UNIFIED IDEOGRAPH-7A74	Hani	LTR	0	4				
喵	U+55B5	CJK UNIFIED IDEOGRAPH-55B5	Hani	LTR	0	5				

A⿺辶⿱穴喵

Drop some fonts here!

Supported formats:
- .designspace, .ufo, .ufoz
- .ttf, .otf, .woff, .woff2, .ttx
- .ttc, .otc

From the 3rd picture, I insert an "A" before U+2FFA, which makes it has a character without the script property Zyyy or Zinh before it, and then, its script property is no longer inherited from the character after it(i.e. Hani) but the character before it(i.e. Latn). U+2FF1 is the same case as in the 2nd picture.

In the 2nd picture, undoubtedly, the sequence ⿺辶⿱穴喵 will form a ligature so long as the font has a "ccmp" feature for this sequence under the Hani script; but, in the 3rd picture, it will never form a ligature no matter what feature the font has, because the sequence mixed the characters with different script property up.

Note that, unless the characters have a script property Zzzz(aka Unknown, i.e. reserved codepoints or PUA characters) or Zxxx, the feature under the DFLT script in the font will NOT be applied to the sequences according to the specification. Of course you can rewrite a Shaping Engine to support the features cross the different script properties, but that actually violates the rule, and lack of standardization.

In brief, a sequence started with a character whose script property is Zyyy is not suitable enough for the encoding model, even if we consider forming a ligature with the IDS itself is legal, as long as a character whose script property is not Hani appears before the first IDC in the sequence, the sequence will no longer ligate.

That is why a starter with its script property Hani should be introduced.

3. What will it do?

Actually, it is used for forming obligatory ligatures, so which sequence could be ligated should be registered to the Unicode Standard, as the IVD registration system, or the Emoji Sequences. Or it will be a bottomless pit with lots of indiscriminate uses. For example, we cannot apply the skin color onto the dog with the unregistered Emoji Sequence, even the characters of the dog and the skin color both exist. It is same that, we cannot use the sequence U+16FE5(HLS) U+2FF0(⿰) U+53E3(口) U+82D7(苗) in order to express U+55B5(喵) forcibly.

Not only the ligature ⿱蟲鳥 can use this encoding model to standardize, many other Han Ligatures could also use it, e.g. the common used 合字 such as 招財進寶, 黄金萬兩, et cetera, or the 反切字 in 同文韻統. Note that U+9FD8..U+9FE9 have the similar case as the 反切字. These are encoded separately due to the historical issue like 𡉏 and 𡋤 mentioned at the beginning of this proposal, but the left ones should never encode as separate characters again.

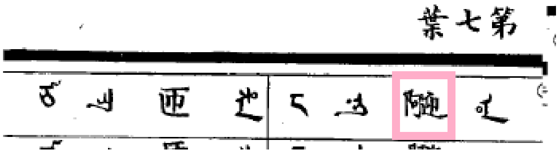


Fig.1 同文韻統第二卷第七葉, ⿰阿迎 for /ŋa/

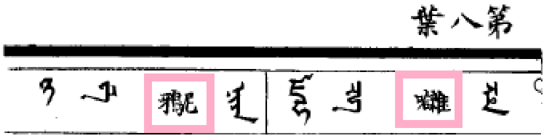


Fig.2 同文韻統第二卷第八葉, ⿰哈雜 for /ʥʱa/ and ⿰鴉尼 for /ɲa/

For example, ⿰鴉尼 will be encoded as the sequence U+16FE5(HLS) U+2FF0(⿰) U+9D09(鴉) U+5C3C(尼) in this encoding model.

After adding the HLS into the Unicode Standard, we will no longer worry about either "the Han Ligatures are unable to input" or "the Han Ligatures waste a lot of codepoints". If this encoding model is accepted, we can even discuss whether to use it to express the Daoist characters or not.

(End of document)